SESSION X MACINTOSH TECHNIQUES

David Lane, *Chair*

PsychLib: A library of machine language routines for controlling psychology experiments on the Apple Macintosh computer

DAVID M. LANE and BRAD ASHBY
Rice University, Houston, Texas

The hardware and system software of the Apple Macintosh computer are described, and their implications for conducting psychological research is discussed. A library of machine language routines for presenting auditory and visual stimuli and recording response latencies is described.

Recently, a new generation of personal computers has been introduced having features that make them particularly well suited for use in psychological research. These features include high-performance microprocessors, large amounts of memory, bit-mapped displays, and high-quality sound generation. The Apple Macintosh, the Atari ST, and the Commodore Amiga are examples of such computers. This paper focuses on using the Apple Macintosh computer in psychological research. An overview of the hardware and system software of this computer is given, followed by a description of a library of machine language subroutines that we developed to facilitate using these features in the control of psychological experiments.

THE MACINTOSH HARDWARE

Overview

The basic hardware of the Macintosh consists of a 7.8-MHz, 68000-cpu, 512 × 342 pixel bit-mapped CRT; two serial ports; an audio port with an associated built-in speaker; and a mouse. It also contains several hundred subroutines in ROM for graphics, sound generation, the keyboard, mouse actions, and other system tasks. The Macintosh can be obtained with either 512K or 1 MB of memory. All Macintoshes now come with an 800K double-sided 3.5-in. disk drive, although earlier models were equipped with a 400K single-sided 3.5-in. drive. These differences in the Macintosh models have no bearing on the following discussion.

Display

The screen of the Macintosh computer is a 9-in.-diagonal white-phosphor CRT. The display is memory mapped with each pixel displayed having a corresponding bit in memory and occupying approximately 1/74 sq in. The display is refreshed at a rate of 60.15 Hz. The vertical refresh interval is thus 1/60.15, or 16.625 msec, and is referred to as a "tick." Interrupts on the vertical refresh and on the horizontal refresh are generated and can be used for software timing. The graphics routines in the Macintosh ROM are collectively referred to as quickdraw. Quickdraw includes routines for drawing, framing, and filling such diverse objects as rectangles, ovals, and arcs, as well as arbitrary regions of the screen.

A series of quickdraw calls may be saved as a "picture" and redrawn at a later time. This provides a standard method for transferring graphics between Macintosh programs and allows the researcher to generate stimuli using any one of several graphics editors, such as MacPaint, or a video digitizer for later presentation under program control.

Sound Generation

Sound generation is performed primarily by the Versatile Interface Adaptor (VIA) chip in the Macintosh. This chip performs many other tasks, including such house-keeping functions as disk motor control, keyboard communication, and monitoring the mouse button.

The ROM routines for sound generation are referred to as the *sound driver*. The sound driver allows three methods for generating sound. The first method generates square waves and thus can be used to produce simple tones of different pitches and volume. The sound is specified as a series of integer triplets, and generating it

The authors' mailing address is: Psychology Department, Rice University, P.O. Box 1892, Houston, TX 77251.

requires approximately 2% of the processor's time. The first integer specifies the frequency, the second the amplitude, and the third the duration. The second method plays sound represented as a series of bytes specifying the waveform of the sound and requires approximately 20% of the processor's time. The third method can produce up to four tones simultaneously, with each tone having separate frequency, phase, and waveform specifications. This last method requires 50% of the processor's time.

The sound driver fetches a byte from the sound buffer in memory every 45 μ sec and uses it as an amplitude value to send to the speaker and audio port. The VIA signals the Macintosh when the sound buffer needs refreshing with new values. The VIA also controls the volume of the generated sound.

Several inexpensive sound digitizers and editors that allow the researcher to capture and manipulate auditory stimuli are commercially available. Most of these editors allow special effects, such as systematic degradation, to be added to the sounds. These stimuli can then be presented later under program control. Gibson (1987) described these sound editors in detail.

Keyboard and Mouse Events

The Event Manager is a collection of routines responsible for handling keyboard and mouse actions, as well as other "events" (e.g., network events). The basic structure of a Macintosh program is to have one big loop that gets an event and then processes it. The ROM routine for obtaining an event is named `GetNextEvent`. As events occur, they are placed in a first-in-first-out queue. `GetNextEvent` retrieves the next event in that queue. As the events are posted in the queue, the time of their occurrence is also posted. The time is reported in ticks since system startup, with one tick occurring every vertical refresh interrupt. (Recall that a tick is 16.625 msec.)

The keyboard is connected to the Macintosh through its own serial interface and contains its own microprocessor. The Macintosh initiates all communication with the keyboard. Normal keyboard communication consists of the Macintosh sending an Inquiry command over the serial line and the keyboard responding with a key transition code. The time required for the sequence beginning with the keypress and ending with the reception of a key transition code is about 6 msec.

The mouse generates interrupts when it moves. The mouse button is connected to a single line of a memory-mapped parallel port of the VIA. Mouse button events are posted every vertical refresh interrupt (mouse-down and mouse-up events are considered separate events). The status of the mouse button can be checked at any time by examining the corresponding bit of the memory-mapped port.

Implications for Recording Response Latency

Since the screen is refreshed every tick, commands to draw on the screen do not have an effect until the next

refresh occurs. Therefore, a stimulus might not be presented for up to 16 msec following the command to present it. This problem can be avoided if the program synchronizes presentation with the vertical refresh by checking an address in low memory.

Every 45 μ sec a bit in the VIA changes from 0 to 1 and then back again. Accurate timing can be obtained by monitoring this bit. If the keyboard is used, 6 msec should be subtracted to adjust for the fact that 6 msec is the time that elapses between a keypress and the occurrence of the interrupt on the Macintosh.

An alternative is to time in terms of ticks. This is very easy to do since the ROM routine `GetNextEvent` returns the time in ticks when the event occurred. Since ticks occur every 16.625 msec, subtracting half a tick (an additional 6 msec if the response is from the keyboard) from the time actually observed results in an unbiased estimate of reaction time that is accurate to within ± 8.3 msec. Although it is possible to time more accurately, this degree of inaccuracy poses no practical problem for most reaction time experiments, even experiments in which the expected effect of the manipulated variable is less than the inaccuracy of 8 msec. Although somewhat paradoxical, this is true because reaction times are averaged over many responses per subject and several subjects per condition and the standard deviation of a timing error of ± 8.3125 msec is small compared with the standard deviation of human reaction time. This is true for even the simplest situations, such as simple reaction times, when the human standard deviation is smallest. Lincoln and Lane (1980) analyzed the decrease in power due to the increase in error variance associated with such a hardware timing error. Using a worst case situation of a one-way within-subjects design with two treatment levels, 9 subjects, and a single reaction time for each subject, they reported that the loss in power at the .05 level for a simple reaction time experiment would drop from .62 to .60. In most instances, the small loss in power due to timing errors could be more than compensated for by just increasing the number of subjects by 1 or by slightly increasing the number of observations per subject.

PSYCHLIB

Microsoft BASIC for the Macintosh provides a convenient and flexible way of calling machine language subroutines. Machine language routines stored in a "Library" can be used to extend the language with each routine adding a statement to the language. These statements can then be used in the same way as any other BASIC statements. For example, if `PsychLib` is a file containing the machine language subroutine named "Bls," then the following statements constitute a valid BASIC program:

Library "PsychLib"

'the following statement is from `PsychLib`

Bls 'black out the screen

PsychLib is a library of machine language subroutines that facilitates writing BASIC programs for the control of psychological experiments on the Macintosh computer.

The main routine for controlling experiments is called SPART (Stimulus Presenter and Response Timer). This routine has the capability of presenting visual and auditory stimuli. To use SPART, one programs the visual and auditory channels separately. For the visual channel, one specifies the sequence of stimuli to be presented. For each stimulus, the location at which it is to be presented, the interstimulus interval (ISI) between the stimulus and the next stimulus, and a bit image of the stimulus are specified. By specifying zero as the ISI, two or more stimuli can be displayed simultaneously. If, for example, one wanted a stimulus to be masked after 50 msec, then the stimulus would be specified with an ISI of 50 msec. The mask would then be presented in the same location as the stimulus thus replacing it.

The auditory channel is programmed separately, making it possible to synchronize auditory and visual presentation. This makes it easy to program such experiments as those in which reaction times to auditory probes occurring at various points in the visual task are required. Both square-wave and free-form sounds can be generated.

The SPART routine can be instructed to return after a specified number of keypresses or to play out its full sequence of stimuli before returning. For each keyboard response, the time and the key pressed is returned. Timing is accurate to ± 1 msec unless free-form sound is used, in which case it is accurate to within 1 tick. For mouse

button pushes, the time (in ticks) and location of the mouse at the time of the button press is returned.

If no sounds are presented, 37% of the screen can be displayed in one refresh cycle. However, it is easy to present four small stimuli at the four corners of the screen in one cycle by specifying ISIs of zero. Although playing simple tones (using square waves) does not significantly slow down the presentation rate (36% of the screen can be displayed in one cycle), free-form sound slows it down so that only 30% of the screen can be displayed in one cycle.

In addition to SPART, PsychLib contains other routines that can be useful for programming experiments. One such routine, RanPer, produces a random permutation of the numbers stored in an integer array. The routine WaitTic waits a specified number of ticks and allows synchronization with the vertical retrace interval. Finally, Bls blacks out the entire Macintosh screen.

Availability

PsychLib is contained on the Rice Psychology Project Disk. This disk can be obtained for \$15 from the Psychology Department, Rice University, P.O. Box 1892, Houston, TX 77251.

REFERENCES

- GIBSON, J. M. (1987). Editing digitized sounds on the Macintosh computer. *Behavior, Research Methods, Instruments, & Computers*, 19, 000-000.
- LINCOLN, C. E., & LANE, D. M. (1980). Reaction time measurement errors resulting from the use of CRT displays. *Behavior Research Methods & Instrumentation*, 12, 55-57.